highlighted and displayed in place of the synchronized revision. A conflict indicator is displayed on the master page of the shared file. The conflict page is displayed alongside the master page when the conflict indicator is selected. The user is presented with both the synchronized state of the master page and the corresponding conflict page. The user may reconcile and merge the conflicting revisions into the master page. Conflicting revisions that are identified as irrelevant may be purged.

Illustrative Operating Environment

With reference to FIG. **1**, one example system for implementing the invention includes a computing device, such as computing device **100**. Computing device **100** may be configured as a client, a server, a mobile device, or any other computing device that interacts with data in a network based collaboration system. In a very basic configuration, computing device **100** typically includes at least one processing unit **102** and system memory **104**. Depending on the exact configuration and type of computing device, system memory **104** may be volatile (such as RAM), non-volatile (such as ROM, flash memory, etc.) or some combination of the two. System memory **104** typically includes an operating system **105**, one or more applications **106**, and may include program data **107**. A revision synchronization module **108**, which is described in detail below, is implemented within applications **106**.

Computing device **100** may have additional features or functionality. For example, computing device **100** may also include additional data storage devices (removable and/or non-removable) such as, for example, magnetic disks, optical disks, or tape. Such additional storage is illustrated in FIG. **1** by removable storage **109** and non-removable storage **110**. Computer storage media may include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information, such as computer readable instructions, data structures, program modules, or other data. System memory **104**, removable storage **109** and non-removable storage **110** are all examples of computer storage media. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computing device **100**. Any such computer storage media may be part of device **100**. Computing device **100** may also have input device(s) **112** such as keyboard, mouse, pen, voice input device, touch input device, etc. Output device(s) **114** such as a display, speakers, printer, etc. may also be included.

Computing device **100** also contains communication connections **116** that allow the device to communicate with other computing devices **118**, such as over a network. Networks include local area networks and wide area networks, as well as other large scale networks including, but not limited to, intranets and extranets. Communication connection **116** is one example of communication media. Communication media may typically be embodied by computer readable instructions, data structures, program modules, or other data in a modulated data signal, such as a carrier wave or other transport mechanism, and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. The term computer readable media as used herein includes both storage media and communication media.

Synchronizing Multiple User Revisions to a Shared File

FIG. **2** illustrates a block diagram of a system for synchronizing multiple user revisions to a shared object. The object may be any entity capable of being shared such as a file. The system includes clients **200**, **210**, **220**, **230**, an email server with file storage capability such as exchange server **240**, web server **250**, peer-to-peer network **260** and email attachment **270**. Clients **200**, **210** are coupled to exchange server **240**. Clients **210**, **220** are coupled to web server **250**. Clients **210**, **220** are also coupled together through peer-to-peer network **260**. Email attachment **270** is arranged to be transferred to and from client **230** by web server **250**. Clients **200**, **210** are both associated with the same user (User **1**). For example, client **200** is accessed by User **1** at home, and client **210** is accessed by User **1** at work. Clients **220**, **230** are associated with different users (User **2** and User **3**, respectively). Clients **200**, **210**, **220**, **230** each include cache **202**, **212**, **222**, **232** for locally storing a shared object. Peer-to-peer network **260** includes virtual server **262** for transferring a shared object between clients **210**, **220**. Revision file **242** and shared objects **252**, **264**, **272** are stored in exchange server **240**, web server **250**, virtual server **262** and email attachment **270**, respectively. Revision file **242** and shared objects **252**, **264**, **272** may be associated with a peer group identifier. The peer group identifier identifies the users who are authorized to access and revise a particular shared object (i.e., the peer group). In one embodiment, the peer group identifier is a uniform resource locator (URL) to the peer group that may be resolved to any web client. Shared objects **252**, **264** are associated with manifest files **254**, **266**, respectively.

Many different users may access, edit and update the same shared object simultaneously through several different transports. For example, User **1** at client **210** and User **2** at client **220**. may access shared object **252** from web server **250**. The shared object is stored locally in corresponding cache **212**, **222**. Both User **1** and User **2** may revise shared object **252**. The revisions are synchronized with shared object **252** on web server **250** such that User **1** can see the revisions made by User **2**, and User **2** can see the revisions made by User **1**.

In another example, User **3** may share access of shared object **272** with User **2** through email attachment **270**. User **2** may revise a locally stored shared object and send an email message to User **3** with the entire shared object or just the revisions to the shared object attached. The revisions made by User **2** are synchronized with shared object **252** on web server **250**. When the email is received at client **230**, the revisions made by User **2** are automatically synchronized with the local shared object stored in cache **232**. User **3** may then make further revisions to shared object **272** and reply to User **2** with the entire shared object or just the revisions to the shared object included as email attachment **270**. The revisions made by User **3** are synchronized with shared object **252** on web server **250**. The shared object at client **220** is also updated to include the revisions made by User **3**.

In another example, User **1** may access a shared object either at home on client **200** or at work on client **210** through exchange server **240**. Exchange servers are often utilized when access to an external server is not permitted or not available. Revision file **242** includes revisions to the shared object. Revision file **242** may be transferred between clients **200**, **210** through a universal serial bus (USB) drive, an email application, or some other mechanism that allows revisions to be transferred back and forth. The revisions are applied to